Routing game in hybrid wireless mesh networks with selfish mesh clients

Yu Wang* and Teresa A. Dahlberg

Department of Computer Science,
University of North Carolina at Charlotte,
Charlotte, NC 28223, USA
E-mail: yu.wang@uncc.edu
E-mail: teresa.dahlberg@uncc.edu
*Corresponding author

Weizhao Wang

Google Inc.,
Mountain View, CA, USA
E-mail: weizhao@google.com

Abstract: Wireless mesh networks (WMNs) consist of mesh routers and mesh clients where fixed mesh routers form the multi-hop backbone of the network. It is often assumed that each individual mesh client will faithfully follow the prescribed protocols. However, these mobile devices, owned by individual users, will likely do what is the most beneficial to their owners, i.e. act 'selfishly'. In this article, we study how to design routing protocols in WMNs with selfish mesh clients. We first show that the totalpayment of the classical Vickrey–Clarke–Groves (VCG)-based routing protocol could be very expensive and inefficient for hybrid mesh networks. Then, we modify the VCG-based method to make it more efficient in terms of total payment, but we also prove that mesh clients could lie about their costs in this modified method. Instead of the VCG-based method, we then propose a novel routing protocol that could achieve Nash equilibrium with low total payments.

Keywords: game theory; Nash equilibrium; routing; VCG mechanism; Vickrey–Clarke–Groves mechanism; wireless mesh networks.

Reference to this paper should be made as follows: Wang, Y., Dahlberg, T.A. and Wang, W. (2008) 'Routing game in hybrid wireless mesh networks with selfish mesh clients', *Int. J. Autonomous and Adaptive Communications Systems*, Vol. 1, No. 4, pp.425–446.

Biographical notes: Yu Wang received his PhD in Computer Science from Illinois Institute of Technology in 2004, his BE and ME in Computer Science from Tsinghua University, China, in 1998 and 2000, respectively. He has been an Assistant Professor of Computer Science at the University of North Carolina at Charlotte since 2004. His current research interests include wireless networks, *ad hoc* and sensor networks, mobile computing, and algorithm design. Wang is a recipient of Ralph E. Powe Junior Faculty Enhancement Awards from Oak Ridge Associated Universities. He is a Member of the *ACM*, *IEEE* and *IEEE Communications Society*.

Teresa A. Dahlberg is an Associate Professor of Computer Science and the Director of the Diversity in Information Technology Institute at the University of North Carolina at Charlotte. She received her BS in Electrical Engineering from the University of Pittsburgh in 1984 and worked as a Hardware and Software Development Engineer for the IBM Corporation for ten years. She received her MS and PhD in Computer Engineering from North Carolina State University in 1990 and 1993, respectively, and joined the faculty at the UNC Charlotte in 1994. Her research interests include routing, quality of service and resource management for mobile networks.

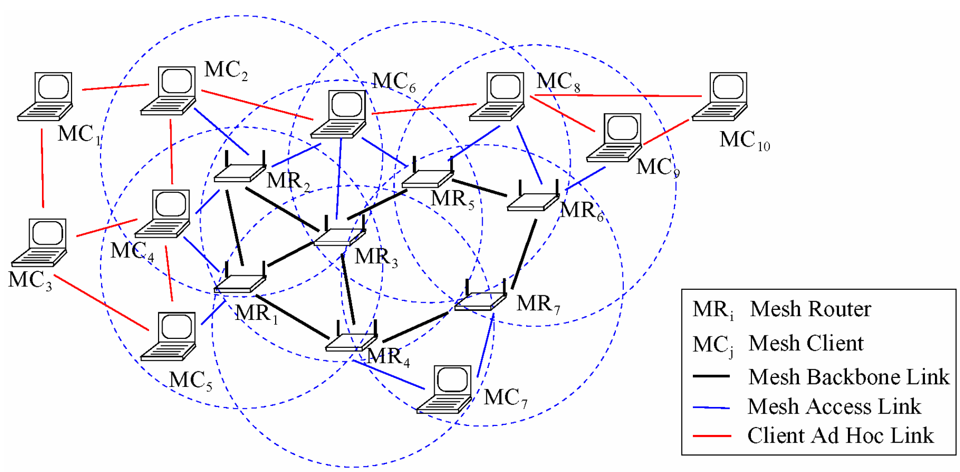
Weizhao Wang received his BS and MS in Computer Science from Shanghai Jiaotong University, China, in 1999 and 2002, respectively, and his PhD in Computer Science from Illinois Institute of Technology in 2006. Currently, he is with Google Inc. His current research interests include wireless networks, game theory, algorithm design and next generation internet.

1 Introduction

Wireless mesh networks (WMNs; Bruno, Conti and Gregori, 2005; Ian, Akyildiz and Wang, 2005) have drawn lots of attention in recent years due to their various potential applications, such as broadband home networking, community and neighbourhood networks, and enterprise networking. They have also been used as the last mile solution for extending the internet connectivity to mobile nodes. Many cities and wireless companies have already deployed WMNs around the world. WMNs consist of two types of nodes: mesh routers and mesh clients. Fixed mesh routers form the multi-hop wireless backbone of the network, while mobile mesh clients can access the network through connections to mesh routers. These networks behave almost like wired networks since the backbone has infrequent topology changes, limited node failures, etc.

The architecture of WMNs usually is either infrastructure/backbone-based or hybrid (Bruno, Conti and Gregori, 2005; Ian, Akyildiz and Wang, 2005). In an infrastructure-based mesh network, mesh clients can only connect to the backbone via single-hop wireless links, while in a hybrid mesh network, mesh clients access the network through mesh routers as well as directly meshing with other mesh clients. In other words, the hybrid architecture combines both infrastructure meshing and client meshing and allows multi-hop connections from mesh clients to mesh routers. Figure 1 shows a simple example of a hybrid mesh network, which contains six mesh routers and ten mesh clients. Mesh clients can communicate with other nodes through either the mesh backbone formed by the mesh routers or an *ad hoc* path (relayed by a small number of other mesh clients). For example, if mesh client MC_6 wants to send a message to MC_9, it can either send the message via mesh routers MR_5 and MR_6 in the backbone or use mesh client MC_8 as the relay node to form a peer-to-peer path. The hybrid architecture has several advantages over the pure infrastructure networks and pure *ad hoc* networks (Wu et al., 2001; Luo et al., 2003; Hsieh and Sivakumar, 2004). For example, the energy consumption of wireless nodes can be reduced, since the distance the signal has to cover is smaller. The coverage of the network can be increased due to the extension of *ad hoc* components. In Figure 1, mesh clients MC_1, MC_3 and MC_{10} are outside the transmission ranges of any mesh routers. However, through the *ad hoc* links, they are connected to the backbone. For the same reason, the number of mesh routers needed to cover certain regions can also be smaller than the one used in a pure infrastructure mesh network. Due to these benefits and many potential applications in various areas, hybrid architectures (not only in mesh networks, but also in general wireless networks) have drawn lots of attention in recent years (Wu et al., 2001; Chandra, Fetzer and Hogstedt, 2002; Dousse, Thiran and Hasler, 2002; Luo et al., 2003; Carbunar, Ioannidis and Nita-Rotaru, 2004; Hsieh and Sivakumar, 2004). Most previous work in hybrid networks deals with various important problems such as routing, quality of service, security, power management, and traffic and mobility modelling. However, there are still many challenges left.

Figure 1 An example of hybrid wireless mesh network (see online version for colours)



In hybrid WMNs, a source node communicates with far off destinations by using mesh routers or intermediate mesh clients as relays. A common assumption made by the majority of the wireless routing protocols is that each wireless node will follow the prescribed protocols without any deviation, e.g. a node is always assumed to relay data packets for other nodes if it is asked to do so by the routing protocols. However, this may not be true in reality: mesh clients could be owned by individual users and thus they will perform in their own interests; the wireless devices are often powered by batteries only. Thus, it is not in the best interest of a wireless node to always forward the data packets for other nodes. If a mesh client refuses to relay the data while the routing protocol assumes that it will, the throughput of the network may decrease and even the network coverage may be hurt. In other words, selfish wireless nodes may hinder the functioning of the network completely. The root cause of the problem is, obviously, that there exist no incentives for mesh clients to be altruistic. Therefore, if we assume that all mesh clients are selfish and wish to maximise their own net gains at all times, providing incentives to them is a must to encourage contribution and thus maintain the robustness and the availability of hybrid WMNs. Thus, a stimulation mechanism is required to encourage the selfish mesh clients to provide service to other clients.

Dealing with selfish users has been well-studied in the economics and networking fields. Recently, there has been a sequence of results (Nisan, 1999; Roughgarden, 2001, 2004; Feigenbaum and Shenker, 2002) published in the theoretical computer science area that tried to solve various problems when the agents are selfish and rational. Here, an agent is rational if it always chooses a strategy that maximises its own gain. A common setting in all these results is that each agent incurs a cost if it is selected to provide the service. For example, in wireless networks, each node will incur an energy cost (and possibly memory cost) when it is asked to relay the data for other nodes. Several protocols (Buttyan and Hubaux, 2000, 2003; Marti et al., 2000; Blazevic et al., 2001; Salem et al., 2003; Srinivasan et al., 2003; Wang, Li and Wang, 2004; Wang et al., 2005, 2006; Zhong et al., 2005; Wang and Li, 2006) have also been proposed recently to address the non-cooperative issue in selfish wireless *ad hoc* networks. Most of these solutions can be applied in WMNs. However, hybrid mesh networks have their own unique characteristics: besides the selfish mesh clients, there are many mesh routers which are fully cooperative, since they belong to the network service provider. In other words, hybrid mesh networks are not purely selfish networks, and the cooperativeness of mesh routers may provide chances of better performance with careful design.

In this article, we study the routing game in a hybrid WMN with selfish mesh clients. We first adapt a VCG-based *ad hoc* routing protocol (Anderegg and Eidenbenz, 2003) for the hybrid mesh network, and show that it is truthful (i.e. nodes reveal their true costs) but could be very expensive in terms of its total payment. Then, we propose a more efficient modified VCG-based protocol in which nodes may lie about their costs but the system will benefit from their lies (i.e. by reducing the total payments). Finally, we also propose a novel efficient routing protocol which could achieve Nash equilibrium with low total payments comparing with VCG-based method. The remainder of this article is organised as follows. First, we introduce some preliminaries (our communication models and the problems to be solved in this article) and related works in Section 2. We study the routing game in hybrid networks in Section 3 (VCG-based) and Section 4 (Nash-equilibrium-based). We conclude our article in Section 5 by pointing out some possible future work.

2 Preliminaries

2.1 Mechanism design

We need to recall a few definitions and concepts from mechanism design. A standard model for mechanism design is as follows. There are n agents $1, \ldots, n$. Each agent i has some private information t_i, called its type, only known to itself. For example, the type t_i can be the cost that agent i incurs for forwarding a packet in a network or can be the maximum payment that the agent i is willing to pay for a service as a service requestor. The agents' types define the type vector $t = (t_1, t_2, \ldots, t_n)$. Each agent i has a set of strategies A_i from which it can choose. For each strategy vector $a = (a_1, \ldots, a_n)$ where agent i plays strategy $a_i \in A_i$, the mechanism $\mathcal{M} = (\mathcal{O}, \mathcal{P})$ computes an output $o = \mathcal{O}(a)$ and a payment vector $\mathcal{P}(a) = (\mathcal{P}_1(a), \ldots, \mathcal{P}_n(a))$. Here, the payment $\mathcal{P}_i(\cdot)$ is the money given to agent i and depends upon the strategies used by the agents.

A valuation function $v(t_i, o)$ assigns a monetary amount to agent i for each possible output o. Let $u_i(t_i, o)$ denote the utility of agent i at the output o of the game, given its type t_i. Here, following a common assumption in the literature, we assume the utility for agent i is quasi-linear, i.e. $u_i(t_i, o) = v(t_i, o) + \mathcal{P}_i(a)$. Let $a_{-i} = (a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n)$ denote the strategies of all the other agents except i. We adopt the assumption in neoclassic economics that every agent is aiming to optimise its utility.

A strategy vector $a^\star$ is called a Nash equilibrium if it maximises the utility of each agent i when the strategies of all the other agents are fixed as $a^\star_{-i}$, i.e. $u_i(t_i,\mathcal{O}(a^\star)) \geq u_i(t_i,\mathcal{O}(a'_i,a^\star_{-i}))$ for all i and all $a'_i \neq a^\star_i$.

A strategy a_i is called a dominant strategy for agent i if it maximises its utility regardless of possible strategies of the other agents, i.e. $u_i(t_i,\mathcal{O}(a_i,b_{-i})) \geq u_i(t_i,\mathcal{O}(a'_i,b_{-i}))$ for all $a'_i \neq a_i$ and all strategies b_{-i} of the agents other than i. If $a = (a_1, a_2, \ldots, a_n)$ is a dominant strategy vector for agents, then a is also a Nash Equilibrium. However, it does not hold for the other side. Thus, Nash Equilibria usually can achieve broader results due to their weaker requirement, while at the same time Nash Equilibria are not as steady as dominant strategies.

A direct-revelation mechanism is a mechanism in which the only actions available to each agent are to report its private type either truthfully or falsely to the mechanism. A direct-revelation mechanism is incentive compatible if reporting valuation truthfully is a dominant strategy. Another very common requirement in the literature for mechanism design is called individual rationality: the agent's utility of participating in the output of the mechanism is not less than the utility of the agent if it did not participate at all. A direct-revelation mechanism is called truthful or strategy-proof if it satisfies both the incentive compatible and individual rationality properties.

The generalised VCG mechanisms by Vickrey (1961), Clarke (1971) and Groves (1973) may be arguably the most important positive result in mechanism design. An objective function is called utilitarian if it is $g(o,t) = \sum_i v(t_i,o)$. The VCG mechanisms apply to (affine) maximisation problems where the objective function is utilitarian and the set of possible outputs is finite. A direct-revelation mechanism $\mathcal{M} = (\mathcal{O}, \mathcal{P})$ belongs to the VCG family if

1 the output $\mathcal{O}(t)$ computed based on the type vector t maximises the utilitarian objective function

2 the payment to agent i is $\mathcal{P}_i(t) = \sum_{j\neq i} v_j(t_j,\mathcal{O}(t)) + h^i(t_{-i})$.

Here, $h^i(\cdot)$ is an arbitrary function of t_{-i}. Green and Laffont (1977) proved that, under mild assumptions, the VCG mechanisms are the only truthful mechanism for utilitarian maximisation problems.

Notice that strategy-proof mechanism always pays a certain amount that is not less than the actual cost to induce the truthfulness. The difference between the total amount paid and the total cost the agents should spend is usually called the premium or overpayment.

2.2 Network models and assumptions

In this article, we consider a hybrid wireless network that consists of n^{MR} mesh routers V^{MR} and n^{MC} mobile mesh clients V^{MC}. We assume each device has an unadjustable transmission range and each communication link is symmetric. Then the communication graph is represented by an undirected graph $G = (V, E)$ where $V = V^{\mathrm{MR}} + V^{\mathrm{MC}} = \{v_1, v_2, \ldots, v_n\}$ is the set of network devices (including both mesh routers and clients), $E = \{e_1, e_2, \ldots, e_m\}$ is the set of wireless links in which a link $e_k = v_iv_j$ means that terminals v_i and v_j can communicate with each other directly. Here, m is the total number of communication links in the graph G and $n = n^{\mathrm{MR}} + n^{\mathrm{MC}}$ is the total number of wireless nodes in the graph. For each node in the mesh network, it has a cost to relay a message. Since we assume fixed transmission range, the cost of a node v_i is a constant, denoted by c_i. Different nodes can have different relay costs. For example, the same type of mesh routers may have a similar cost but various mesh clients (i.e. laptops, PDAs and cell phones) should have different transmission costs. Let $\mathbf{c} = (c_1, c_2, \ldots, c_n)$ be the cost vector of all nodes (including mesh clients and mesh routers).

The following assumptions are adopted in this article:

1. Each mobile device (acting as a relay node) is selfish and rational: it tries to maximise its own benefit.
2. The cost to relay transit traffic for each node is private information; it is only known by the node itself by considering all facts which may affect its own benefit (such as remaining power, transmission range, environment, etc.).
3. Wireless mesh routers are cooperative, i.e. they will reveal their costs[1] truthfully, since they are owned by the network service provider.
4. The source of the routing will pay the selected relay nodes.
5. The wireless nodes do not collude to improve their benefit.
6. The network is bi-connected, which implies that if we remove the agent the network is still connected.

The last assumption is necessary to prevent some nodes from being a monopoly and charging arbitrary costs, and to increase network robustness.

2.3 Problem statement

We are now ready to describe our routing scenario in more detail. Without loss of generality, we assume that s is the source, t is the destination and s and t are in V. Remember that there is a fixed cost c_i for each node v_i to transmit a unit size data and the cost to transmit traffic of size h is $h \times c_i$. Every wireless node is required to declare a cost d_i for forwarding the unit size data. The set of all declared costs is denoted as $\mathbf{d} = (d_1, \ldots, d_n)$. Notice that d_i may not equal c_i, which is v_i's actual cost. Let $\mathbf{d}|^k d_k' = (d_1, \ldots, d_k - 1, d_k', d_k + 1, \ldots, d_n)$ which is the declared cost with a different declared cost d_k' for v_k. For simplicity of our analysis, we normalise the traffic to unit size data. Dropping this assumption does not change the results. Our aim is to design unicast routing mechanisms.

Definition 1. A unicast routing mechanism $\mathcal{M} = (\mathcal{O}, \mathcal{P})$ consists of a routing output function $\mathcal{O}$ and a payment function $\mathcal{P}$. Let $\mathbf{d}$ be the vector of declarations by all relay agents. $\mathcal{O}(\mathbf{d}) = (\mathcal{O}_1(\mathbf{d}), \ldots, \mathcal{O}_n(\mathbf{d}))$ is an output function vector such that $\mathcal{O}_i(\mathbf{d})$ indicates whether the node v_i should send the packet, i.e. $\mathcal{O}_i(\mathbf{d}) = 1$ indicates that node v_i should forward the packet, i.e. participate in the route. $\mathcal{P}(\mathbf{d}) = (\mathcal{P}_1(\mathbf{d}), \ldots, \mathcal{P}_n(\mathbf{d}))$ is a payment function vector that computes the payment for the terminals, i.e. $\mathcal{P}_i(\mathbf{d})$ is payment to terminal v_i.

The routing scenario is as follows. The routing algorithm selects a route P (such as the least cost path) according to the declared cost vector $\mathbf{d}$. To stimulate cooperation among all wireless nodes, the source s pays the selected relaying nodes a certain amount of money for forwarding the data. For each node v_i, the source s computes a payment $\mathcal{P}_i(\mathbf{d})$ according to the declared cost vector $\mathbf{d}$. The utility, in standard economic model, of node v_i is $u_i(\mathbf{d}) = \mathcal{P}_i(\mathbf{d}) - \mathcal{O}_i(\mathbf{d})\cdot c_i$, where $\mathcal{O}_i(\mathbf{d})$ indicates whether v_i relays the packet. We always assume that the wireless nodes are rational: node v_i always tries to maximise its utility $u_i(\mathbf{d})$.

If the routing algorithm and the payment scheme are not well-designed, a node v_j may improve its utility by lying about its cost, i.e. by declaring a cost d_j such that $d_j \neq c_j$. The objective of this article is then to design a unicast routing mechanism such that each node v_j has to declare its true cost, i.e. $d_j = c_j$, to maximise its utility either with fixed strategies of other nodes or regardless of possible strategies of the other nodes. In other words, we want to design routing protocols for a selfish WMN that can achieve a dominant strategy (satisfying incentive compatibility). In addition, we also hope the designed mechanisms satisfy the following two properties:

1 *Individual rationality*. A mobile device is guaranteed to have non-negative utility if it reports its type truthfully no matter what other devices do.

2 *Polynomial time computability*. All computations (the computation of the output and the payment) are done in polynomial time.

Notations. In the following, we introduce some terminology and symbols that will be used later. If a mobile node v_s wants to send data to another mobile device v_t, typically, the path with the minimum sum of relaying cost (least cost path or shortest path) is used to route the packets. The shortest path between two nodes v_s, v_t under declared cost vector $\mathbf{d}$ is denoted as $\text{LCP}(v_s, v_t, \mathbf{d})$. Among all the paths between v_s, v_t, with node v_k on it, the shortest path is denoted as $\text{LCP}_{v_k}(v_s, v_t, \mathbf{d})$. Similarly, among all paths between s, t without node v_k on it, the shortest path is denoted as $\text{LCP}_{-v_k}(v_s, v_t, \mathbf{d})$ or $\text{LCP}(v_s, v_t, \mathbf{d}|^k \infty)$. Given a simple path P in the network G with cost vector $\mathbf{d}$, the sum[2] of the relaying cost of nodes on path P is denoted as $|\text{P}(\mathbf{d})|$. Let $\mathcal{P}(\mathbf{d})$ be the total payment to the terminals, i.e. $\mathcal{P}(\mathbf{d}) = \sum_i \mathcal{P}_i(\mathbf{d})$. Here, $\mathcal{P}_i(\mathbf{d})$ (or $\mathcal{P}_i(v_s, v_t, \mathbf{d})$) is the payment of node v_i for relaying the packet from v_s to v_t under declared cost vector $\mathbf{d}$. For notational simplicity, we also use $\mathcal{O}(\mathbf{d})$ to denote the node set that is selected to route the data. The source always pays the total payment to $\mathcal{O}(\mathbf{d})$.

2.4 Priori arts

The most well-known and widely used incentive-based method to solve the non-cooperative problem is the VCG mechanism family by Vickrey (1961), Clarke (1971) and Groves (1973). Several mechanisms (Nisan and Ronen, 1999; Feigenbaum et al., 2002; Anderegg and Eidenbenz, 2003), which essentially belong to the VCG mechanism family, have been proposed in the literature to assure the cooperation for routing problems in a general network.

Routing has been an important part of the algorithmic mechanism-design from the very beginning. Nisan and Ronen (1999) provided a polynomial-time strategy-proof mechanism for optimal unicast route selection in a centralised computational model. In their formulation, the network is modelled as an abstract graph $G = (V, E)$. Each edge e of the graph is an agent and has a private type t_e, which represents the cost of sending a message along this edge. The mechanism-design goal is to find a least cost path LCP(x, y, **d**) between two designated nodes x and y. The valuation of an agent e is $-t_e$ if the edge e is the part of the path LCP(x, y, **d**) and 0 otherwise. Nisan and Ronen used the VCG mechanism for payment.

Feigenbaum et al. (2002) then addressed the truthful low cost routing in a different network model. They assume that each node k incurs a transit cost c_k for each transit packet it carries. For any two nodes i and j of the network, $T_{i,j}$ is the intensity of the traffic (number of packets) originating from i and destined for node j. Their strategy-proof mechanism again is essentially the VCG mechanism. They gave a distributed method such that each node i can compute a payment $P_k(i, j, \mathbf{d}) > 0$ to node k for carrying the transit traffic from node i to node j if node k is on the LCP(i, j, **d**). Anderegg and Eidenbenz (2003) proposed a similar routing protocol for wireless *ad hoc* networks based on VCG mechanism again. They assumed that each link has a cost and each node is a selfish agent. Recently, Wang and Li (2006) proposed a time optimal method to compute the VCG payment in a centralised manner and studied in detail how to implement the routing protocol in the distributed manner for *ad hoc* networks.

There is a vast literature on the mechanism design or implementation paradigm in which some mechanisms are designed to achieve the socially desirable outcomes in spite of users' selfishness. Some of these approaches use Nash equilibrium rather than dominant-strategy (VCG uses dominant-strategy). That is, they assumed that simultaneous selfish play leads to a self-consistent Nash equilibrium in which no agent can improve its utility by deviating from its current strategy when other agents keep their strategies. Notice that since Nash equilibrium has a weak requirement on the strategies used by the agents, it often can achieve a much wider variety of outcomes.

How to achieve cooperation among selfish terminals in networks was previously addressed in Buttyan and Hubaux (2000, 2003), Marti et al. (2000), Blazevic et al. (2001), Srinivasan et al. (2002, 2003) and Jakobsson, Hubaux and Buttyan (2003), In Marti et al. (2000), nodes which agree to relay traffic but do not, are termed as misbehaving. Their protocol avoids routing through these misbehaving nodes. In Buttyan and Hubaux (2000, 2003), Blazevic et al. (2001) and Jakobsson, Hubaux and Buttyan (2003), a secure mechanism to stimulate nodes to cooperate is presented. The key idea behind these approaches is that terminals providing a service should be remunerated, while terminals receiving a service should be charged. Each terminal maintains a counter, called nuglet counter, in a tamper resistant hardware module, which is decreased when the terminal sends a packet as originator and increased when the terminal forwards a packet. Both these methods belong to so called credit-based method. Usually, they are heuristics and need some special hardware. In recent years, incentive-based methods have been proposed to solve the non-cooperative problem. The most well-known and widely used incentive-based method is the VCG mechanism family by Vickrey (1961), Clarke (1971) and Groves (1973). Several mechanisms (Nisan and Ronen, 1999; Feigenbaum et al., 2002; Anderegg and Eidenbenz, 2003) which essentially belong to the VCG mechanism family have been proposed in the literature to assure the cooperation for the unicast problem in a general network.

Unlike in *ad hoc* networks, not much research has been conducted for non-cooperative problems in wireless hybrid networks until recently. Lamparter, Paul and Westhoff (2003) proposed a charging scheme to motivate cooperation in hybrid networks. They assume the existence of an internet service provider (ISP) that authenticates the nodes involved in a given communication and takes care of charging or rewarding them. The amount of charging or rewarding is chosen by the ISP and not related to the cost of the relay node. In Salem et al. (2003) and Jakobsson, Hubaux and Buttyan (2003), the authors also proposed an incentive mechanism that is based on charging/rewarding scheme and makes collaboration rational for selfish nodes. However, the amount of charging or rewarding is also decided as a fixed rate. Weyland, Staub and Braun (2005) also proposed incentive-based schemes called CASHnet in which the payment or reward is based on the distance to the gateway. Not all these charging/rewarding methods are truthful for selfish participants.

3 VCG-based truthful routing

3.1 Truthful VCG-based payment scheme

Our VCG-based payment scheme is similar to the *ad hoc* one by Anderegg and Eidenbenz (2003), except

1 our network model is a node-based model where each node is the selfish agent

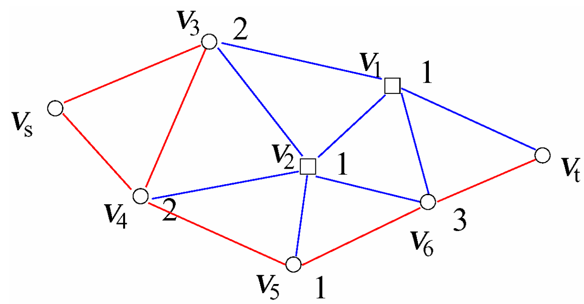
2 we have some special handling with mesh routers since they are cooperative.

Assume that the node v_s has to send a packet to v_t through the relay of some other nodes. It pays these relay nodes to compensate their costs for carrying the transit traffic incurred by v_s. Here, the payment for each node (or user) chould be in any format, such as virtual currency, furure usage time, or other incentives. The output of the algorithm is the path connecting v_s and v_t with the minimum cost, which is known as LCP(v_s, v_t, **d**). The payment scheme is based on the VCG. The payment for node v_k is

$$\mathcal{P}_k(\mathbf{d}) = \begin{cases} 0, & \text{if } v_k \notin \mathrm{LCP}(v_s, v_t, \mathbf{d}) \\ d_k, & \text{if } v_k \in \mathrm{LCP}(v_s, v_t, \mathbf{d}) \cap V^{\mathrm{MR}} \\ |\,\mathrm{LCP}_{-v_k}(s, t, \mathbf{d})\,| - |\,\mathrm{LCP}(v_s, v_t, \mathbf{d})\,| + d_k, & \text{if } v_k \in \mathrm{LCP}(v_s, v_t, \mathbf{d}) \cap V^{\mathrm{MC}}. \end{cases} \quad (1)$$

Here, $\mathrm{LCP}_{-v_k}(v_s, v_t, \mathbf{d})$ denotes the least cost path between nodes v_s and v_t if we remove node v_k and all its adjacent links from the original graph. Notice that for a mesh router v_k in LCP(v_s, v_t, **d**), the payment is just its claim cost d_k, since we always assume the mesh router is cooperative, i.e. it always claims its true costs.

Figure 2 An example of hybrid wireless mesh networks (see online version for colours)



For example, assume we want to send a unit packet from node v_s to node v_t in Figure 2. Hereafter, we always use squares to represent mesh routers and use circles to represent mesh clients in all figures. The number beside the square/circle is its cost. In the network, there are two mesh routers v_1 and v_2. The most cost-efficient path is

$$\text{LCP}(v_s, v_t, \mathbf{d}) = v_s, v_3, v_1, v_t$$

with $|\text{LCP}(v_s, v_t, \mathbf{d})| = 2 + 1 = 3$. Since v_1 is a mesh router, its payment is its claim cost (also true cost), i.e.

$$\mathcal{P}_1(\mathbf{d}) = 1.$$

For mesh client v_3, we need to calculate its payment. The shortest path without node v_3 is

$$\text{LCP}_{-v_3}(v_s, v_t, \mathbf{d}) = v_s, v_4, v_2, v_1, v_t$$

with $|\text{LCP}_{-v_3}(v_s, v_t, \mathbf{d})| = 2+1+1 = 4$. Thus, we have the VCG payment for node v_3

$$\mathcal{P}_3(\mathbf{d}) = 4 - 3 + 1 = 3.$$

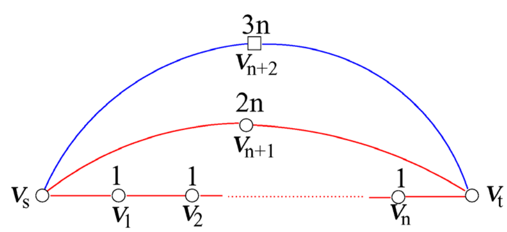
This payment falls into the VCG mechanism, so it is strategy-proof. In other words, the mechanism satisfies the incentive compatibility property: if $d_k = c_k$, node v_k maximises its utility $\mathcal{P}_k(\mathbf{d}) - x_k \cdot c_k$. Here, $x_k = 1$ if $v_k \in \text{LCP}(v_s, v_t, \mathbf{d})$, otherwise $x_k = 0$. In addition, it is not hard to prove that the other two properties: individual rationality and polynomial time computability. All the proofs are similar to those in Nisan and Ronen (1999) and Anderegg and Eidenbenz (2003).

3.2 *Inefficiency of payments*

As discussed in Immorlica et al. (2005), the VCG mechanism has the attractive property that each node's dominant strategy is to reveal its true cost. However, the VCG mechanism can lead to the customer paying far more than the true cost of the cheapest path. The overpayment is because a bonus needs to be paid to every relay node on the path. Remember in Equation (1), node v_s pays each mobile node v_k on $\text{LCP}(v_s, v_t, \mathbf{d})$ more than its actual cost to make sure that it will not lie about its cost. The overpaid value is the improvement of the LCP due to the existence of node v_k. It is not difficult to construct a network example such that the over-payment of a node v_k could be arbitrarily large. This is extraordinarily inefficient for wireless networks. Consider the network shown in

Figure 3. The nodes v_i for $1 \le i \le n+1$ are the mobile devices as selfish mesh clients while v_{n+2} is a mesh router. Assume that we want to send packets from node v_s to v_t. The least cost path is $v_s, v_1, v_2, \ldots, v_n, v_t$ with cost n. Using the VCG mechanism, the payment to node v_i is $\mathcal{P}_i(\mathbf{d}) = |\mathrm{LCP}_{-v_i}(v_s, v_t, \mathbf{d})| - |\mathrm{LCP}(v_s, v_t, \mathbf{d})| + d_i = 2n - n + 1 = n + 1$. Notice that path $\mathrm{LCP}_{-v_i}(v_s, v_t, \mathbf{d})$ is the path v_s, v_{n+1}, v_t whose cost is $2n$. Therefore, the total payment of the least cost path will be $n(n+1)$, which is $n+1$ times the true cost of the path. Thus, we say that the VCG-method is an expensive method since the overpayment ratio can be arbitrarily large. Besides the overpayment problem, in some cases, the payment to the lowest-cost path in the VCG-method may even greatly exceed the true cost of the second-cheapest path. Consider again the example in Figure 3. The total VCG payment of the selected path is $(n+1)/2$ times the payment for the path v_s, v_{n+1}, v_t, while the cost of the second shortest path is only two times that of the shortest path. In addition, notice that since the only relaying node in path v_s, v_{n+2}, v_t is mesh router v_{n+2} who must claim its true cost. In other words, if we choose the path v_s, v_{n+2}, v_t, then it only costs $3n$, while the VCG method selects the least cost path and spends $n(n+1)$. This is not reasonable for the source node or the system. Thus, we want to investigate new methods to fix this problem.

Figure 3 Link's paradox for VCG-based unicast (see online version for colours)



3.3 *Modified VCG-based routing*

It is very natural for us to make the following modification: after applying the VCG-method, we check whether there exists a path from the source to the target using only mesh routers as the relaying nodes; if one exists, we compare the true cost of this path with the total payments of the path that the VCG method selects, and use the cheaper one as the final routing path. The detailed modified algorithm is given in Algorithm 1.

Algorithm 1 Modified VCG-based Routuing scheme

1. Every node v_k claims its cost d_k. Let $\mathbf{d} = (d_1, \ldots, d_n)$ be the all claimed costs for all n nodes in G.
2. Compute the least cost path $\mathrm{LCP}(v_s, v_t, \mathbf{d})$ from v_s to v_t in graph G and the least cost path $\mathrm{LCP}_{-v_k}(v_s, v_t, d)$ if we remove node v_k and all its adjacent links from the original graph.
3. Compute the payments for the least cost path as follows. If $v_k \in \mathrm{LCP}(v_s, v_t, \mathbf{d})$ and is a mesh router, $\mathcal{P}_k(\mathbf{d}) = d_k$. If $v_k \in \mathrm{LCP}(v_s, v_t, \mathbf{d})$ and is a mesh client,
   $\mathcal{P}_k(\mathbf{d}) = |\mathrm{LCP}_{-v_k}(v_s, v_t, \mathbf{d})| - |\mathrm{LCP}(v_s, v_t, \mathbf{d})| + d_k$. If $v_k \notin \mathrm{LCP}(v_s, v_t, \mathbf{d})$, $\mathcal{P}_k(\mathbf{d}) = 0$.

4 Check whether there exists a path $P'(v_s, v_t, \mathbf{d})$ between v_s and v_t using mesh routers as all the relaying nodes. If this kind of path does not exist, output the path $\text{LCP}(v_s, v_t, \mathbf{d})$ and pay each node v_k on the path $\mathcal{P}_k(\mathbf{d})$. Exit.

5 If there exist paths that all relaying nodes are mesh routers, we select the path $P'(v_s, v_t, \mathbf{d})$ with the least cost among these access point paths, and calculate the total cost of this path $|P'(v_s, v_t, \mathbf{d})| = \sum_{v_i \in P'(v_s, v_t, \mathbf{d})} d_i$. In addition, calculate the VCG payment $\mathcal{P}(v_s, v_t, \mathbf{d}) = \sum_{v_k \in \text{LCP}(v_s, v_t, \mathbf{d})} \mathcal{P}_k(\mathbf{d})$. If $\mathcal{P}(v_s, v_t, \mathbf{d}) \le |P'(v_s, v_t, \mathbf{d})|$, output the path $\text{LCP}(v_s, v_t, \mathbf{d})$ and pay each node v_k on the path $\mathcal{P}_k(\mathbf{d})$. Otherwise, output the path $P'(v_s, v_t, \mathbf{d})$ and pay each node v_k its claimed cost d_k.

Notice that the modified VCG-method is not a real VCG method anymore, since the output changes (maybe not the optimum path in terms of total cost). Unlike the VCG method, we can prove that the modified VCG method is not truthful.

Theorem 1. The modified VCG method is not truthful.

Proof. Naturally, one conjecture that some relay nodes on the shortest path $\text{LCP}(v_s, v_t, \mathbf{c})$ but not selected due to the large payment may have incentive to lie down their costs in order to reduce the payment. However, the following example shows the contrary. In the same network shown in Figure 3, we select path v_s, v_{n+2}, v_t instead of the shortest path $v_s, v_1, \ldots, v_n, v_t$ under the modified VCG-method, since it only uses a mesh router as a reply node and has less cost than the VCG-payment. Now considering the scenario when node v_1 reveals its cost as $n + 1 - \varepsilon$ for a small positive ε. The shortest path does not change, and VCG-payment to node v_i is $1 + \varepsilon$ for $2 \le i \le n$. The total VCG-payment of the path $v_s, v_1, \ldots, v_n, v_t$ is $(n-1)\cdot(1+\varepsilon)+(n+1) = 2n+(n-1)\cdot\varepsilon = 2n+1$ when $\varepsilon = 1/(n-1)$. This is smaller than the cost $3n$ of v_s, v_{n+2}, v_t. Thus, the modified VCG-method will choose path $v_s, v_1, \ldots, v_n, v_t$ with relay nodes instead of the pure mesh router v_{n+2}. Therefore, node v_1 will benefit from cheating. □

However, from the system or the source node view, the total payment of the routing decreases, and therefore it is more cost-efficient. We can have the following observation: under the modified VCG method, the relay nodes only have incentives to lie about their costs upward and the resulting total payment is not greater than the total payment in the VCG method. In other words, this kind of cheating is good cheating. Moreover, this case happens only when a cheap pure mesh router path exists.

In Algorithm 1, we consider only the pure mesh router paths from s to t as the alternative solution to reduce the total cost. We can then generalise the idea by considering all partial mesh router paths to reduce further the overall payment. The refined method is given in Algorithm 2.

Algorithm 2 *New modified VCG-based routing scheme*

1 Every node v_k claims its cost d_k. Let $\mathbf{d} = (d_1, \ldots, d_\text{n})$ be the all claimed costs for all n nodes in G.

2 Build the virtual graph G' only composed of mesh routers V^{MR} and the source v_s and the target v_t. Thus, G' has $n^{\text{MR}} + 2$ nodes. Each mesh router has its cost as the node cost.

3 **for** each pair of nodes v_i and v_j in G' **do**

4   **if** v_i and v_j are connected in G **then**

5   Add a link v_iv_j in G'.

6   **else**

7   Compute the least cost path LCP(v_i, v_j, **d**) from v_i to v_j in graph G and compute the total VCG-payment to the relay agents on LCP(v_i, v_j, **d**), say $\mathcal{P}(v_i, v_j, \mathbf{d})$.

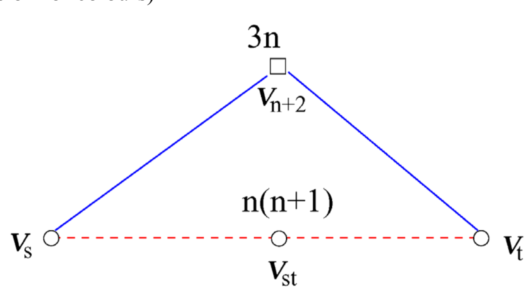
8   Add a virtual link v_iv_j with a virtual node v_{ij} seats in the middle in G′, and set the node cost of v_{ij} as $\mathcal{P}(v_i, v_j, \mathbf{d})$.

9 After above loop, G' is a complete graph on V^{MR} and v_s and v_t. Find the least cost path in the virtual graph G' from v_s to v_t and output the union of the corresponding paths in the original graph G.

10 Every mesh router in the output gets a payment equal to its cost; every relay agent in the output gets its own VCG-payment on the virtual link.

The basic idea of the above method is to build a virtual complete graph G' on all mesh routers and v_s and v_t. Two nodes v_i and v_j (they are mesh routers or v_s or v_t) are connected by an actual link in G', if they are connected in G. Otherwise, a virtual link between v_i and v_j will be added in G', and this link represents the least cost path LCP(v_i, v_j, **d**) in G. To have a cost of this virtual link, a virtual node v_{ij} with a cost of the total VCG-payment for this path is added. The algorithm will output the least cost path from v_s and v_t in G'. Figure 4 shows the virtual graph constructed for the example network in Figure 3. Links v_sv_{n+2} and $v_{n+2}v_t$ are directly links from G among mesh routers and v_s and v_t, while link v_sv_t is a virtual link of the least cost path LCP(v_s, v_t, **d**) in G. v_{st} is a virtual node with the total VCG-payment of LCP(v_s, v_t, **d**) as its cost. The algorithm will output the path $v_sv_{n+2}v_t$ since it has less cost than the other path in G'. Notice that in this special case, Algorithm 2 has the same output with Algorithm 1. However, in general hybrid networks with more mesh routers, it will have a different output with Algorithm 1 and will usually have a smaller total payment. It is easy to show that this algorithm is also not truthful but the overpayment is much less than the VCG-method.

Figure 4 Virtual graph G' constructed in Algorithm 1 for the example network in Figure 3 (see online version for colours)



4 Nash-equilibrium-based routing

In light of the potential large overpayment of the VCG mechanism which is the only strategy-proof mechanism if the path selected is the least cost path, it is natural to relax the dominant strategy to the Nash equilibrium, which is a weaker requirement. In this section, we study how to design a mechanism for hybrid networks that can induce some Nash Equilibria. However, Immorlica et al. (2005) showed that if we simply pay whatever the node reports, there does not exist any Nash Equilibrium. Due to the non-existence of the Nash Equilibrium, they introduced the concept of a strong ε-Nash equilibrium, in which there is no group of agents who can deviate in a way that improves the payoff of each member by at least ε. Then, they proposed a modified first price auction for link-weighted networks that can achieve ε-Nash Equilibrium. With further modification of the auction, we obtain our new routing method that induces efficient Nash Equilibria.

Algorithm 3 presents our new routing algorithm. In our method, each node declares two bids instead of one: the first bid vector $\mathbf{d}$ is used to compute the least cost path, the second bid vector $\mathbf{d}'$ is used to decide payment for the nodes on the least cost path. The method can be divided into two phases. In the first phase, each node sends a packet that either contains its declared cost or some other information. Then, the mesh network gives a small 'bonus' to each node such that each node v_i maximises its utility when it bids its true cost no matter what other nodes do. In the second phase, we first choose the least cost path LCP(v_s, v_t, $\mathbf{d}$) using the first bid vector $\mathbf{d}$. Then, each node v_i on this least cost path bids a second bid d_i'. We obtain a new cost vector $\mathbf{h}$ from $\mathbf{d}$ and $\mathbf{d}'$ by setting $h_i = d_i$ if $v_i \notin$ LCP(v_s, v_t, $\mathbf{d}$) and $h_i = d_i'$ if $v_i \in$ LCP(v_s, v_t, $\mathbf{d}$). Now we compute the path P = LCP(v_s, v_t, $\mathbf{h}$), and each node on P should relay the packet and receive a payment h_i. Last but not the least important, the nodes that are in LCP(v_s, v_t, $\mathbf{d}$) but not in P receive a fine $\gamma \mid d_i' - d_i \mid$ for being too greedy in bidding d'_i.

Algorithm 3. Nash-eqilibrium-based routing

1 Every node v_i claims two costs d_i and d_i'. Notice that if v_i is a mesh router, both its claimed costs are equal to its true cost. Let $\tilde{\mathbf{d}} = \langle \mathbf{d}, \mathbf{d}' \rangle$ be the declared cost vector for all nodes.

First phase

2 For each node $v_i \in G$, set $f_i(v_s, v_t, \mathbf{d}) = \tau_i(d_{-i})\left[d_u\left(n \cdot d_u - \sum_{v_j \in G - v_i} d_j\right) - d_i^2 / 2\right]$.

Here, d_u is the maximum cost any node can declare and $\tau_i(d_{-i})$ is any function that does not depend on d_i.

3 Every node sends a dummy packet of size $\rho = \tau_i(d_{-i})\left(n \cdot d_u - \sum_{v_i \in G} d_i\right)$ which may contain any arbitrary data.

4 Compute the path $\tilde{P} = \text{LCP}(v_s, v_t, \mathbf{d})$. For each node v_i on $\tilde{P}$, set $h_i = d'_i$, set $h_i = d_i$ for other nodes.

Second phase

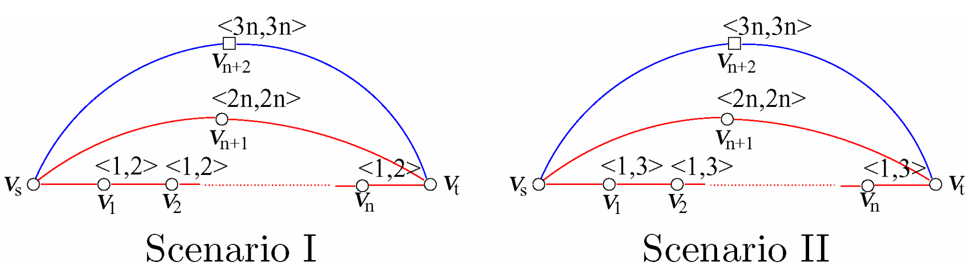
5 Compute path LCP(v_s, v_t, **h**) and break ties according to the following rule: if two paths have the same cost, then we choose the one that contains more nodes from path $\tilde{P}$.

6 Output the path LCP(v_s, v_t, **h**) as the final relay path. For any node v_i on LCP(v_s, v_t, **h**), the payment $\mathcal{P}_i(v_s, v_t, \tilde{\mathbf{d}}) = f_i(v_s, v_t, \mathbf{d}) + h_i$. For each node v_i on LCP(v_s, v_t, **d**) but not on LCP(v_s, v_t, **h**), the payment $\mathcal{P}_i(v_s, v_t, \tilde{\mathbf{d}}) = f_i(v_s, v_t, \mathbf{d}) - \gamma \,|\, d'_i - d_i \,|$. For other node v_i, the payment $\mathcal{P}_i(v_s, v_t, \tilde{\mathbf{d}}) = f_i(v_s, v_t, \mathbf{d})$.

To show how our Nash-equilibrium-based method works, we consider the same network in Figure 3 and give two example bidding scenarios shown in Figure 5. The two numbers in the bracket at node v_i show the bids d_i and d'_i. In Scenario I, LCP(v_s, v_t, **d**) is $v_s v_1 \ldots v_n v_t$. Thus, $h_i = d'_i = 2$ for $1 \le i \le n$. Note that under cost vector **h**, both paths $v_s v_{n+1} v_t$ and $v_s v_1 \ldots v_n v_t$ have the same cost $2n$. However, according to the tie-breaking rule, we will choose $v_s v_1 \ldots v_n v_t$ because it contains more nodes from LCP(v_s, v_t, **d**). Thus, v_i for $1 \le i \le n$ gets 2 in the second phase. In Scenario II, LCP(v_s, v_t, **d**) is also $v_s v_1 \ldots v_n v_t$. Similarly, $h_i = d'_i = 3$ for $1 \le i \le n$. However, path $v_s v_1 \ldots v_n v_t$ has cost $3n$ in the cost vector **h**, which is greater than the cost of path $v_s v_{n+1} v_t$. Thus, all nodes v_i for $1 \le i \le n$ are punished by fine -2γ, respectively. Node v_{n+1} gets a payment $2n$ in the second phase, and the actual routing path is $v_s v_{n+1} v_t$.

Regarding the first phase in Algorithm 3, we can prove the following lemma which shows each node does not have incentive to lie about its cost.

Figure 5 Illustration of Nash-equilibrium-based method (see online version for colours)



Lemma 2. For each node v_i, the utility in the first phase $g_i(\mathbf{d}) = -\rho \cdot c_i + f_i(v_s, v_t, \mathbf{d})$, strictly decreases in $[c_i, +\infty)$ and strictly increases in $(-\infty, c_i]$ on d_i.

Proof. The utility $g_i(\mathbf{d})$ can be simplified as follows.

$$\begin{aligned} g_i(\mathbf{d}) &= -\rho \cdot c_i + f_i(v_s, v_t, \mathbf{d}) \\ &= \tau \cdot \left[d_u \cdot (n \cdot d_u - d_{-v_i}) - \frac{d_i^2}{2} \right] - c_i \cdot \tau \cdot \left(n \cdot d_u - \sum_{v_i \in G} d_i \right) \\ &= \tau \left[(d_u - c_i) \cdot (n \cdot d_u - d_{-v_i}) - \frac{d_i^2}{2} + c_i \cdot d_i \right] \\ &= \tau_i(d_{-i}) \left[(d_u - c_i) \cdot (n \cdot d_u - d_{-v_i}) + \frac{c_i^2}{2} - \frac{(c_i - d_i)^2}{2} \right], \end{aligned}$$

where $d_{-v_i} = \sum_{v_j \in G - v_i} d_j$. Notice that $\tau_i(d_{-i})[(d_u - c_i) \cdot (n \cdot d_u - d_{-v_i}) + (c_i^2 / 2)]$ does not depend upon d_i. Thus, $g_i(\mathbf{d})$ if a function on d_i that decreases in $[c_i, +\infty)$ and increases in $(-\infty, c_i]$. □

We now prove that there must exist some Nash equilibrium for our Nash-equilibrium-based method.

Theorem 3. There exists Nash Equilibrium for our Nash-equilibrium-based method.

Proof. We prove by explicitly constructing a bid vector $\langle c_i, c'_i \rangle$, an Nash Equilibrium for our method. Without loss of generality, we assume that P = LCP(v_s, v_t, $\mathbf{c}$) = $v_s v_1 v_2 \ldots v_k v_t$. Recall that $\mathbf{c}$ is the true cost vector. We initialise the cost vector $\mathbf{c}^{(0)} = \mathbf{c}$ and iteratively process the nodes from v_1 to v_k as follows: compute VCG payment $\mathcal{P}_i^{\text{VCG}}\left(v_s, v_t, \mathbf{c}^{(i-1)}\right)$ for v_i using the cost vector $\mathbf{c}^{(i-1)}$, and obtain the new cost vector $\mathbf{c}^{(i)}$ from $\mathbf{c}^{(i-1)}$ by setting $c_i^{(i)} = \mathcal{P}_i^{\text{VCG}}\left(v_s, v_t, \mathbf{c}^{(i-1)}\right)$. Let $\mathbf{c}' = \mathbf{c}^{(k)}$ and the bidding vector $\tilde{\mathbf{d}} = \langle \mathbf{c}, \mathbf{c}' \rangle$.

We now prove that the bidding vector $\tilde{\mathbf{d}} = \langle \mathbf{c}, \mathbf{c}' \rangle$ is a Nash Equilibrium by contradiction. For the sake of contradiction, we assume that node v_i can increase its utility by declaring a bid pair $\langle x, y \rangle$ that is different from $\langle c_i, c'_i \rangle$. We discuss by cases.

Case 1. Node $v_i \in$ LCP(v_s, v_t, $\mathbf{c}$). There are two subcase here.

1 If $v_i \in$ LCP(v_s, v_t, $\mathbf{c}|^i x$), i.e. v_i is the LCP generated in the first phase, then the utility in the first phase is $g_i(\mathbf{c}|^i x) \le g_i(\mathbf{c})$ from Lemma 2. In the second phase, c'_i is the maximum value v_i can declare when it is still in LCP(v_s, v_t, $\mathbf{c}'$). Thus, the utility of v_i does not increase.

2 If $v_i \notin$ LCP(v_s, v_t, $\mathbf{c}|^i x$), i.e. v_i is not in the LCP generated in the first phase, then $x > c_i$. Let P′ ∈ LCP(v_s, v_t, $\mathbf{c}|^i x$) that covers v_i and P ∈ LCP(v_s, v_t, $\mathbf{c}$). Here, P′ is on the path generated in the first phase while P is on the true least cost path. Notice that for any node $v_j \notin$ LCP(v_s, v_t, $\mathbf{c}$), $c'_j = c_j$. Thus, for any node $v_j \in$ P′, $h_j = c'_j = c_j$. On the other hand, since path P is not part of the generated path, $h_j = c_j$ for any node

$v_j \in \mathrm{P}-v_i$. From the assumption P′ is on the generated LCP, $\omega(\mathrm{P}', \mathbf{c}) \le \omega(\mathrm{P}, \mathbf{c}|^i x)$. Here, $\omega(\mathrm{P}, \mathbf{c})$ is the total cost of a path P with cost vector $\mathbf{c}$. Thus,

$$\omega(\mathrm{P}',\mathbf{h}) = \sum_{v_j \in \mathrm{P}'} h_j = \sum_{v_j \in \mathrm{P}'} c_j = \omega(\mathrm{P}',\mathbf{c}) \le \omega(\mathrm{P},\mathbf{c}|^i x) = \sum_{v_j \in \mathrm{P}-v_i} c_j + x$$

$= \sum_{v_j \in \mathrm{P}-v_i} h_j + x = \omega(\mathrm{P},\mathbf{h}|^i x)$ which implies that v_i is not selected in the unicast phase. Therefore, its overall utility is $g_i(\mathbf{c}|^i x)$. Now we conclude that v_i can not increase its utility by declaring a bid pair $\langle x, y\rangle$ that is different from $\langle c_i, c'_i\rangle$ in this case.

Case 2. Node $v_i \notin \mathrm{LCP}(v_s, v_t, \mathbf{c})$. There are also two subcases here.

1 If v_i is not in $\mathrm{LCP}(v_s, v_t, \mathbf{c}|^i x)$, then $\mathbf{h} = \mathbf{c}'|^i x$. v_i's utility from the first phase is $g_i(\mathbf{c}|^i x)$, which is not greater than $g_i(\mathbf{c})$ from Lemma 2. If v_i is not on $\mathrm{LCP}(v_s, v_t, \mathbf{h})$, then its overall utility is $g_i(\mathbf{c}|^i x)$, which is smaller than $g_i(\mathbf{c})$. If v_i is on $\mathrm{LCP}(v_s, v_t, \mathbf{h})$, then $x < c_i$, which means that v_i's utility in the second phase is $x-c_i < 0$. Thus, v_i does not increase its overall utility.

2 If $v_i \in \mathrm{LCP}(v_s, v_t, \mathbf{c}|^i x)$, i.e. v_i manages to decrease its first bid x to be on the generated path, then $x < c_i$. Thus, the utility in the second phase decreases. Let $\mathrm{P} \in \mathrm{LCP}(v_s, v_t, \mathbf{c})$ be a bridge that covers $\mathrm{P}' \in \mathrm{LCP}(v_s, v_t, \mathbf{c}|^i x)$ and $v_i \in \mathrm{P}'$. If $v_i \notin \mathrm{LCP}(v_s, v_t, \mathbf{h})$, then it has utility $-\gamma |d'_i - d_i|$ in the second phase. If $v_i \in \mathrm{LCP}(v_s, v_t, \mathbf{h})$, then $\omega(\mathrm{P}', \mathbf{h}) \le \omega(\mathrm{P}, \mathbf{h})$. Similarly to the argument of case 1, every v_j in P and $\mathrm{P}'-v_i$ has cost c_j. Thus, y must be smaller than c_i. Therefore, v_i's utility in the second phase is $y - c_i < 0$. This shows v_i's overall utility decreases when v_i bids $\langle x, y\rangle$ which is different from $\langle c_i, c'_i\rangle$ in this case. This finishes our proof. □

More generally, if $\tilde{\mathbf{d}}$ is any Nash Equilibrium of our method, we have the following lemma.

Lemma 4. Assume that $\tilde{\mathbf{d}} = \langle \mathbf{d}, \mathbf{d}'\rangle$ is a Nash Equilibrium for our method where $\mathbf{h}$ is the cost vector obtained in Algorithm 3.

1 $\mathbf{d} = \mathbf{c}$, i.e. each node declares its true cost as the first bid.

2 $\mathrm{LCP}(v_s, v_t, \mathbf{c}) = \mathrm{LCP}(v_s, v_t, \mathbf{h})$, i.e. our algorithm always chooses actual least cost path.

3 For any $v_i \in \mathrm{LCP}(v_s, v_t, \mathbf{d}) = \mathrm{LCP}(v_s, v_t, \mathbf{c})$, $|\mathrm{LCP}(v_s, v_t, \mathbf{h})| = |\mathrm{LCP}_{-v_i}(v_s, v_t, \mathbf{h})|$.

Proof. Statement 1. For simplicity for our notation, let $\mathrm{P}_1 = \mathrm{LCP}(v_s, v_t, \mathbf{d})$ and $\mathrm{P}_2 = \mathrm{LCP}(v_s, v_t, \mathbf{h})$. We first prove that $\mathrm{P}_1 = \mathrm{P}_2$. We prove this by studying edges in different groups.

- $v_i \notin \mathrm{P}_1 \cup \mathrm{P}_2$. In this case, v_i's overall utility is $g_i(\mathbf{d})$, which is maximised when $d_i = c_i$. Since, we directly set $h_i = d_i$ for link $v_i \notin \mathrm{P}_1$, $h_i = d_i = c_i$.
- $v_i \in \mathrm{P}_1$ and $v_i \notin \mathrm{P}_2$. In this case, v_i's overall utility is $g_i(\mathbf{d}) - \gamma |d'_i - d_i|$, which is maximised when $d_i = d'_i = c_i$. Thus, $h_i = d'_i = d_i = c_i$.
- $v_i \in \mathrm{P}_2$ and $v_i \notin \mathrm{P}_1$. In this case, we have $h_i = d_i$.

Then, we can conclude that for any node v_j that is not in P_2, $h_j = c_j = d_j$. Thus,

$$\omega(\mathrm{P}_2,\mathbf{h})-\omega(\mathrm{P}_1,\mathbf{h}) = \sum_{v_j\in \mathrm{P}_2-\mathrm{P}_1} h_j - \sum_{v_j\in \mathrm{P}_1-\mathrm{P}_2} h_j$$
$$\geq \sum_{v_j\in \mathrm{P}_2-\mathrm{P}_1} d_j - \sum_{v_j\in \mathrm{P}_1-\mathrm{P}_2} c_j = \sum_{v_j\in \mathrm{P}_2-\mathrm{P}_1} d_j - \sum_{v_j\in \mathrm{P}_1-\mathrm{P}_2} d_j$$
$$=\sum_{v_j\in \mathrm{P}_2} d_j - \sum_{v_j\in \mathrm{P}_1} d_j = \omega(\mathrm{P}_2,\mathbf{d})-\omega(\mathrm{P}_1,\mathbf{d})\geq 0.$$

This implies that $\omega(\mathrm{P}_1, \mathbf{d}) = \omega(\mathrm{P}_2, \mathbf{d})$ and $\omega(\mathrm{P}_1, \mathbf{h}) = \omega(\mathrm{P}_2, \mathbf{h})$. Thus, $\mathrm{P}_1 = \mathrm{P}_2$.

Then we consider all remaining edges, i.e. $v_i \in \mathrm{P}_2 \cap \mathrm{P}_1$. Notice that $\mathrm{P}_2 \cap \mathrm{P}_1 = \mathrm{P}_1$ since $\mathrm{P}_1 = \mathrm{P}_2$. If $d_i \geq c_i$, then by declaring c_i, v_i's utility in the first phase increases. In the meanwhile, v_i is still on P_1, which means that its utility in the second phase does not change. Thus, $d_i \geq c_i$ for each node on P_1. If v_i is on P_1 and $d_i < c_i$, then it can increase its utility from the first phase by bidding $d_i = c_i$. In the meanwhile, $d_i \leq c_i$ for each node v_i on P_1 and $d_i = c_i$ for other nodes. Thus, v_i can guarantee that it is still on the least cost path when it declared c_i. Therefore, v_i can increase its overall utility by bidding $d_i = c_i$, which contradicts the definition of the Nash Equilibrium. This finishes the proof of the first statement.

Statement 2. The second statement is straightforward from the definition.

Statement 3. Assume that $v_i \in \mathrm{LCP}(v_s, v_t, \mathbf{d}) = \mathrm{P}_1$. If $|\mathrm{LCP}(v_s,v_t,\mathbf{h})| < |\mathrm{LCP}_{-v_i}(v_s,v_t,\mathbf{h})|$, then by bidding $\mathbf{d}' + \delta$ for a sufficiently small δ such that $|\mathrm{LCP}(v_s,v_t,\mathbf{h}'|^i\, d_i'+\delta)| < |\mathrm{LCP}_{-v_i}(v_s,v_t,\mathbf{h}')|$, its utility increases by δ. Thus, $|\mathrm{LCP}(v_s,v_t,\mathbf{h})| \geq |\mathrm{LCP}_{-v_i}(v_s,v_t,\mathbf{h})|$. On the other hand, from the definition of the least cost path, $|\mathrm{LCP}(v_s,v_t,\mathbf{h})| \leq |\mathrm{LCP}_{-v_i}(v_s,v_t,\mathbf{h})|$. Thus, $|\mathrm{LCP}(v_s,v_t,\mathbf{h})| = |\mathrm{LCP}_{-v_i}(v_s,v_t,\mathbf{h})|$ for every $v_i \in \mathrm{P}_1$. □

Unlike the VCG mechanism that always has the same total payment for a fixed cost vector **c**, the total payments may vary under different Nash Equilibria for our method. Let $\tilde{\mathbf{d}}^{\min}$ and $\tilde{\mathbf{d}}^{\max}$ be any two Nash Equilibria of our method such that $P(v_s,v_t,\tilde{\mathbf{d}})$ is minimised and maximised respectively. Usually, persons favour a truthful mechanism over the Nash Equilibrium because the system may have multiple Nash Equilibria and it is almost impossible to reach some specific Nash Equilibria in a distributed setting. However, it is possible that the system performance under different Nash Equilibrium does not differ much, and we are not so worried about which Nash Equilibrium the system converges to. Fortunately, our method does have this nice property and the following theorem shows that the total payment at different Nash Equilibrium differs at most two times.

Theorem 5. $P(v_s,v_t,\tilde{\mathbf{d}}^{\max}) \leq 2P(v_s,v_t,\tilde{\mathbf{d}}^{\min})$, i.e. the stability of our method is 2.

Proof. Karlin, Kempe and Tamir (2005) showed that

$$2\cdot \sum_{v_j\in \mathrm{LCP}(v_s,v_t,\mathbf{c})} d'^{\min}_j \geq \sum_{v_j\in \mathrm{LCP}(v_s,v_t,\mathbf{c})} d'^{\max}_j.$$

From Statement 1 of Lemma 4, the total payment for the second phase is a function of $\mathbf{d} = \mathbf{c}$. Thus, as long as the true cost vector $\mathbf{c}$ is fixed, the total payment for the second phase is fixed, say ε. From Statement 2 of Lemma 4, The total payment $P(v_s, v_t, \tilde{\mathbf{d}}) = \sum_{v_j \in \mathrm{LCP}(v_s, v_t, \mathbf{c})} d'_j$ for any Nash Equilibrium $\mathbf{d}$. Thus,

$$\mathcal{P}(v_s, v_t, \tilde{\mathbf{d}}^{\max}) = \sum_{v_j \in \mathrm{LCP}(v_s, v_t, \mathbf{c})} d_j^{\max} + \varepsilon \le 2 \cdot \sum_{v_j \in \mathrm{LCP}(v_s, v_t, \mathbf{c})} d_j^{\min} + 2\varepsilon = 2\mathcal{P}(v_s, v_t, \tilde{\mathbf{d}}^{\min}). \quad \square$$

Another application of the minimum Nash Equilibrium is the notation of frugality Karlin, Kempe and Tamir (2005) that is used to measure the overpayment of any mechanism. The frugality of a mechanism is defined as the total payment over $v(\mathbf{c})$, where $v(\mathbf{c}) = \sum_{v_j \in \mathrm{LCP}(v_s, v_t, \mathbf{c})} d_j^{\min}$. It has been shown by Karlin, Kempe and Tamir (2005) that VCG mechanism has frugality $O(n)$ in unicast routing where n is the number of nodes. On the contrary, our method has a frugality $2 + \varepsilon$ for any given positive ε, which *greatly* improves the VCG mechanism and is asymptotically optimal. Before presenting our theorem, we give a lemma that relates the $v(\mathbf{c})$ to the cost of the least bridge cover $\mathbb{LB}(v_s, v_t, \mathbf{c})$.

Lemma 6. Immorlica et al. (2005) and Karlin, Kempe and Tamir (2005). For any network, $v(\mathbf{c}) \le |\mathbb{LB}(v_s, v_t, \mathbf{c})| \le 2v(\mathbf{c})$.

Theorem 7. The frugality of our method is $2 + \varepsilon$ for any given positive ε, which asymptotically optimal.

Proof. In the proof of Theorem 5, we obtain that

$$2v(\mathbf{c}) = 2 \cdot \sum_{v_j \in \mathrm{LCP}(v_s, v_t, \mathbf{c})} \mathbf{d}'^{\min}_j \ge \sum_{v_j \in \mathrm{LCP}(v_s, v_t, \mathbf{c})} \mathbf{d}'^{\max}_j$$

which implies that total payment in the second phase is at most two times the $v(\mathbf{c})$.

In the first phase, by setting $\tau_i(d_{-i}) = (\varepsilon\, |\mathbb{LB}(v_s, v_t, \mathbf{d})|) / 2n$ for each $v_i \in \mathrm{LCP}(v_s, v_t, \mathbf{d})$ and $\tau_i(d_{-i}) = (\varepsilon\, |\mathrm{LCP}(v_s, v_t, \mathbf{d})|) / n$ otherwise, the total payment in the first phase is at most $\varepsilon \cdot v(\mathbf{c})$ under any Nash Equilibrium $\tilde{\mathbf{d}}$. Thus, the total payment is at most $(2 + \varepsilon)v(\mathbf{c})$, which finishes our proof. $\square$

Theorem 7 reveals an important fact: our method could largely reduce the overpayment of the VCG method.

5 Conclusions

In this article, we studied the routing game in hybrid WMNs with selfish mesh clients. We proposed two routing methods to stimulate cooperation among selfish mesh clients: VCG-based and Nash equilibrium-based. The classical VCG-based routing mechanism is truthful (i.e. each node will declare its true cost and also follow the designed protocol) and easy to implement. However, as all VCG mechanisms, the proposed scheme pays each mesh client more than its declared cost to prevent it from lying, thus the overpayment could be large in the worse case. Since in hybrid mesh networks mesh routers are cooperative, the output of the VCG method is not efficient in terms of total payment. Then by considering the cooperation with mesh routers we proposed two modified VCG methods to refine the efficiency of the VCG method. However, we proved they are not truthful anymore. Finally, we proposed our Nash equilibrium method which adapts a novel idea of double bidding to address the overpayment problem of the VCG-based method. We proved our method could achieve Nash equilibrium with very low total payment compared with VCG-payment. There is still a number of interesting questions left. For example, our protocols assume that nodes will not collude and the cost of a wireless node is fixed. However, in practice, collusion and dynamic cost are very common. Therefore, we leave the study of collusion and dynamic cost model as our future work.

Acknowledgement

The work of Yu Wang was supported, in part, by funds provided by Oak Ridge Associated Universities and the University of North Carolina at Charlotte.

References

Anderegg, L. and Eidenbenz, S. (2003) '*Ad hoc* – VCG: a truthful and cost-efficient routing protocol for mobile *ad hoc* networks with selfish agents', Paper presented in the Proceedings of the *9th ACM Annual International Conference on Mobile Computing and Networking (MobiCom)*, pp.245–259.

Blazevic, L., Buttyan, L., Capkun, S., Giordano, S., Hubaux, J.P. and Boudec, J.Y.L. (2001) 'Self-organization in mobile *ad-hoc* networks: the approach of terminodes', *IEEE Communications Magazine*, Vol. 39, pp.138–147.

Bruno, R., Conti, M. and Gregori, E. (2005) 'Mesh networks: commodity multihop *ad hoc* networks', *IEEE Communications Magazine*, Vol. 43, pp.123–131.

Buttyan, L. and Hubaux, J.P. (2000) 'Enforcing service availability in mobile ad-hoc wans', Paper presented in the Proceedings of the *1st ACM International Symposium on Mobile ad hoc Networking and Computing*, pp.87–96.

Buttyan, L. and Hubaux, J. (2003) 'Stimulating cooperation in self-organizing mobile *ad hoc* networks', *ACM/Kluwer Mobile Networks and Applications*, Vol. 5, pp.579–592

Carbunar, B., Ioannidis, I. and Nita-Rotaru, C. (2004) 'Janus: towards robust and malicious resilient routing in hybrid wireless networks', *WiSe '04: Proceedings of the 2004 ACM Workshop on Wireless Security*, pp.11–20, ACM Press.

Chandra, R., Fetzer, C. and Hogstedt, K. (2002) 'Adaptive topology discovery in hybrid wireless networks', Paper presented in the Proceedings of the *1st International Conference on Ad-Hoc Networks and Wireless (AdHoc-Now)*, Toronto.

Clarke, E.H. (1971) 'Multipart pricing of public goods', *Public Choice*, Vol. 11, pp.17–33.

Dousse, O., Thiran, P. and Hasler, M. (2002) 'Connectivity in ad-hoc and hybrid networks', Paper presented in the Proceedings of the *Annual Joint conference of the IEEE Computer and Communications Societies (INFOCOM)*, New York.

Feigenbaum, J. and Shenker, S. (2002) 'Distributed algorithmic mechanism design: recent results and future directions', *DIALM '02: Proceedings of the 6th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, pp.1–13, ACM Press.

Feigenbaum, J., Papadimitriou, C., Sami, R. and Shenker, S. (2002) 'A BGP-based mechanism for lowest-cost routing', Paper presented in the Proceedings of the *2002 ACM Symposium on Principles of Distributed Computing*, pp.173–182.

Green, J. and Laffont, J.J. (1977) 'Characterization of satisfactory mechanisms for the revelation of preferences for public goods', *Econometrica*, Vol. 45, pp.427–438.

Groves, T. (1973) 'Incentives in teams', *Econometrica*, Vol. 41, pp.617–631.

Hsieh, H-Y. and Sivakumar, R. (2004) 'On using peer-to-peer communication in cellular wireless data networks', *IEEE Transactions on Mobile Computing*, Vol. 3, pp.57–72.

Ian, F., Akyildiz, X.W. and Wang, W. (2005) 'Wireless mesh networks: a survey', *Computer Networks*, Vol. 47, pp.445–487.

Immorlica, N., Karger, D., Nikolova, E. and Sami, R. (2005) 'First-price path auctions', Paper presented in the Proceedings of the *6th ACM Conference on Electronic Commerce(EC'05)*.

Jakobsson, M., Hubaux, J-P. and Buttyan, L. (2003) 'A micro-payment scheme encouraging collaboration in multi-hop cellular networks', Paper presented in the Proceedings of the *Financial Cryptography*.

Karlin, A., Kempe, D. and Tamir, T. (2005) 'Beyond VCG: frugality of truthful mechanisms', Paper presented in the Proceedings of *IEEE Symposium of Foundations of Computer Science (FOCS)*, pp.615–626.

Lamparter, B., Paul, K. and Westhoff, D. (2003) 'Charging support for *ad hoc* stub networks', *Elsevier Computer Communications*, Vol. 26, pp.1504–1514.

Luo, H., Ramjee, R., Sinha, P., Li, L.E. and Lu, S. (2003) 'Ucan: a unified cellular and ad-hoc network architecture', *MobiCom '03: Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, pp.353–367, ACM Press.

Marti, S., Giuli, T.J., Lai, K. and Baker, M. (2000) 'Mitigating routing misbehavior in mobile *ad hoc* networks', Paper presented in the *Proceedings of ACM Annual International conference on Mobile computing and Networking (MobiCom)*.

Nisan, N. (1999) 'Algorithms for selfish agents', *Lecture Notes in Computer Science*, Vol. 1563, pp.1–15.

Nisan, N. and Ronen, A. (1999) 'Algorithmic mechanism design', Paper presented in the Proceeding of the *31st Annual Symposium on Theory of Computing (STOC99)*, pp.129–140.

Roughgarden, T. (2001) 'Designing networks for selfish users is hard', Paper presented in the Proceedings of *IEEE Symposium on Foundations of Computer Science*, pp.472–481.

Roughgarden, T. (2004) *Selfish Routing and the Price of Anarchy*. Cambridge, MA: MIT Press.

Salem, N.B., Buttyan, L., Hubaux, J-P. and Jakobsson, M. (2003) 'A charging and rewarding scheme for packet forwarding in multi-hop cellular networks', *MobiHoc '03: Proceedings of the 4th ACM International Symposium on Mobile ad hoc Networking and Computing*, pp.13–24, ACM Press.

Srinivasan, V., Nuggehalli, P., Chiasserini, C.F. and Rao, R.R. (2002) 'Energy efficiency of *ad hoc* wireless networks with selfish users', Paper presented in the Proceedings of *European Wireless Conference 2002 (EW2002)*.

Srinivasan, V., Nuggehalli, P., Chiasserini, C.F. and Rao, R.R. (2003) 'Cooperation in wireless *ad hoc* networks', Paper presented in the Proceedings of *Annual Joint conference of the IEEE Computer and Communications Societies (INFOCOM)*.

Vickrey, W. (1961) 'Counterspeculation, auctions and competitive sealed tenders', *Journal of Finance*, Vol. 16, pp.8–37.

Wang, W. and Li, X-Y. (2006) 'Low-cost routing in selfish and rational wireless *ad hoc* networks', *IEEE Transactions on Mobile Computing*, Vol. 5, pp.596–607.

Wang, W., Li, X-Y. and Wang, Y. (2004) 'Truthful multicast in selfish wireless networks', Paper presented in the Proceedings of *ACM Annual International Conference on Mobile Computing and Networking (MobiCom 2004)*.

Wang, W., Li, X-Y., Sun, Z. and Wang, Y. (2005) 'Design multicast protocols for non-cooperative networks', Paper presented in the Proceedings of *Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM05)*.

Wang, W., Eidenbenz, S., Wang, Y. and Li, X-Y. (2006) 'OURS: optimal unicast routing system in non-cooperative wireless networks', paper presented in the Proceeding of *ACM Annual International Conference on Mobile Computing and Networking (MobiCom 2006)*.

Weyland, A., Staub, T. and Braun, T. (2005) 'Comparison of incentive-based cooperation strategies for hybrid networks', Paper presented in the Proceedings of the *Third International Conference on Wired/Wireless Internet Communications (WWIC 2005)*.

Wu, H., Qiao, C., De, S. and Tonguz, O. (2001) 'Integrated cellular and *ad hoc* relaying systems: iCAR', *IEEE Journal on Selected Areas in Communications*, Vol. 19, pp.2105–2115.

Zhong, S., Li, L., Liu, Y.G. and Yang, Y. (2005) 'On designing incentive-compatible routing and forwarding protocols in wireless ad-hoc networks: an integrated approach using game theoretical and cryptographic techniques', Paper presented in the Proceedings of the *11th Annual International Conference on Mobile Computing and Networking (ACM Mobicom)*.

Notes

[1] We can set the costs of mesh routers as zeros, if we consider that the mesh clients already pay the service fee to the network service provider. However, in this article, we assume the user still needs to pay the network service provider for particular uses of the mesh routers for relaying messages.

[2] Notice that the costs of *s* and *t* are not included, since all path share them.